

## **Assignment 3: Implementation of Data Model in PostgreSQL**

**Name:** Tomáš Meravý Murárik

**Student ID:** 127232

**Submission Date:** X.X.XXXX

## logical-to-physical mapping of model

### GENERAL CHANGES

-	Created ENUM types: attributes, combat_status, effect_type	Centralized allowed values for critical fields
-	All ID fields implemented as SERIAL	Auto-increment standardized across all tables
-	Added NOT NULL constraints where applicable	Enforced data integrity for required fields
-	Added CHECK constraints	Validations for health $\geq 0$ , stats between 1-99, etc.
-	Created sequences for all ID fields	Explicit sequence control for primary keys

### Character

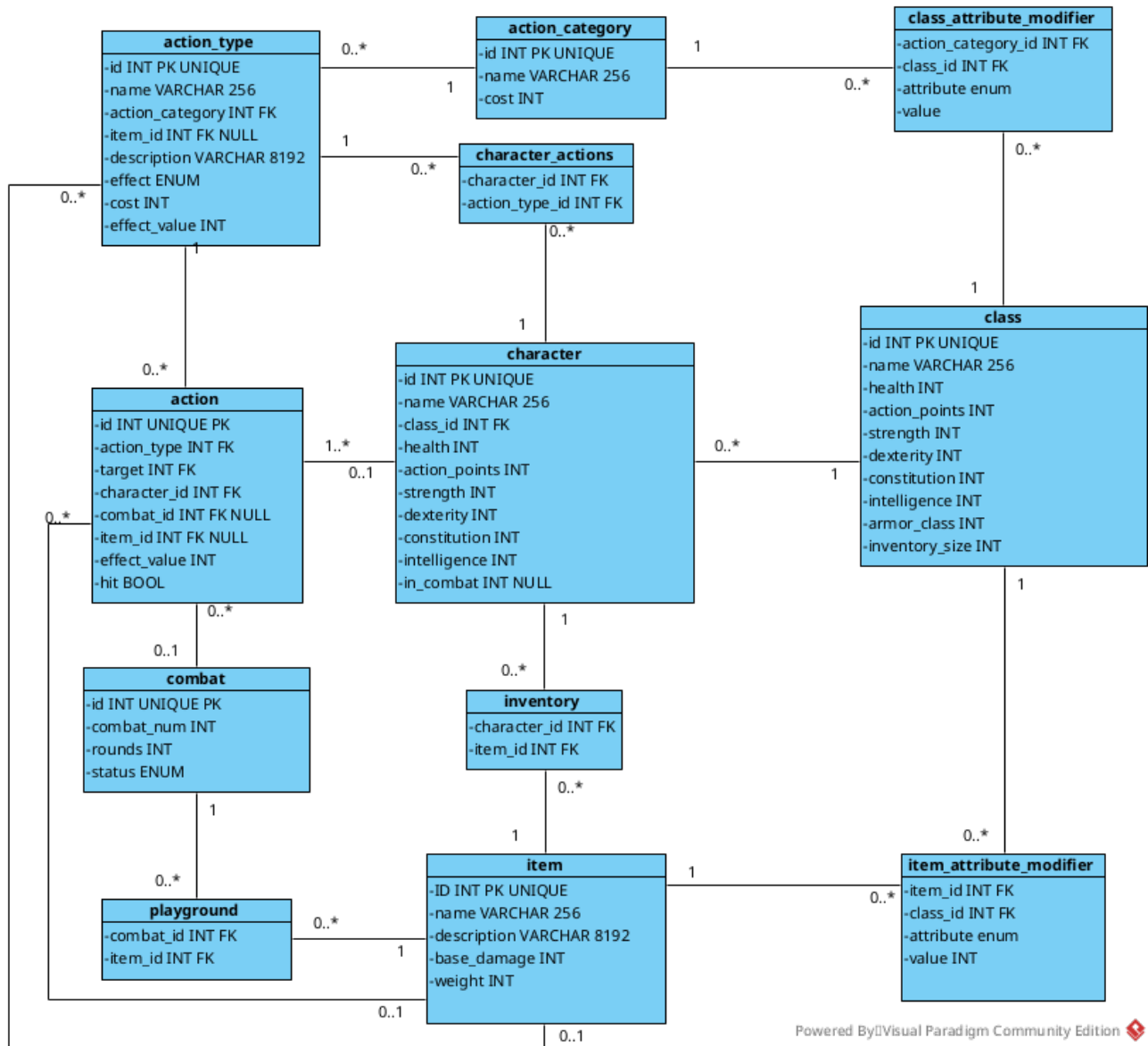
Logical Model	Physical Implementation	Changes/Decisions
id INT PK	id SERIAL PRIMARY KEY	Auto-increment added here and to all other id's
in_combat BOOL	in_combat INT	Changed to store combat_num identifying which combat the character is in(NULL if not in combat)

### item\_attribute\_modifier

Logical Model	Physical Implementation	Changes/Decisions
id INT PK	(No id column)	Removed surrogate key - natural key used (item_id+class_id+attribute)

### Action

Logical Model	Physical Implementation	Changes/Decisions
hit BOOL	hit BOOL NOT NULL	Made NOT NULL for data integrity

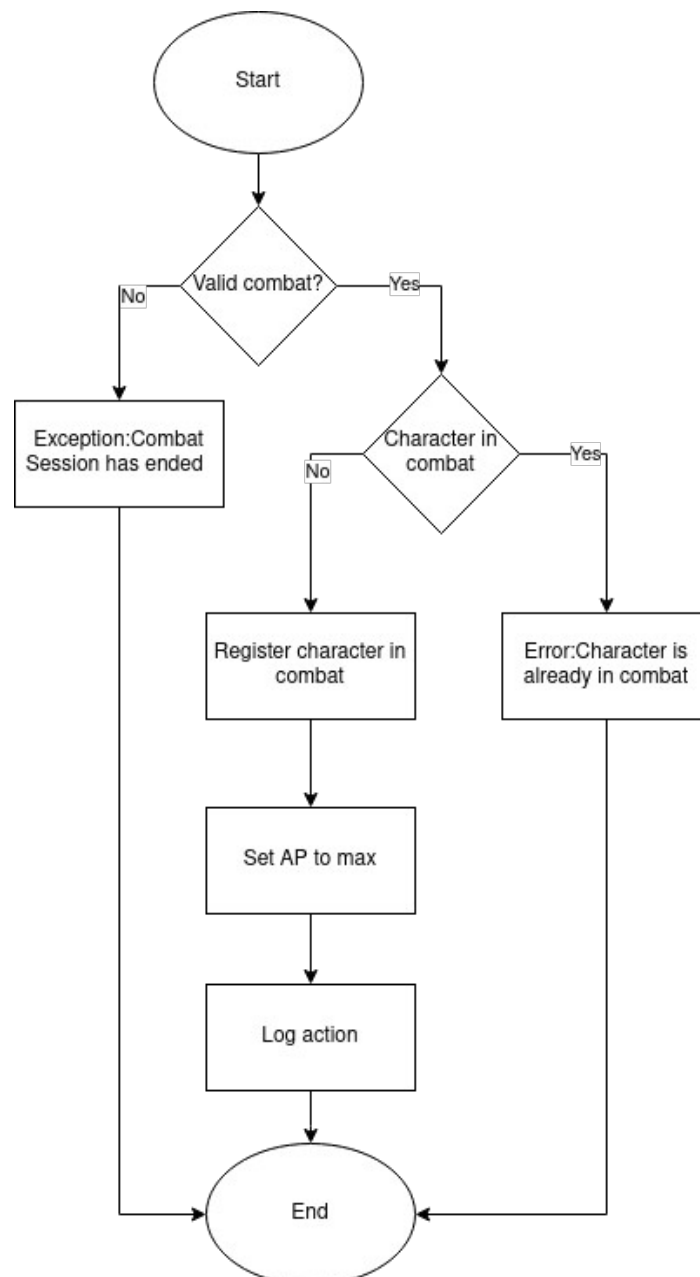


# Combat Flow Overview

## Procedures/Functions:

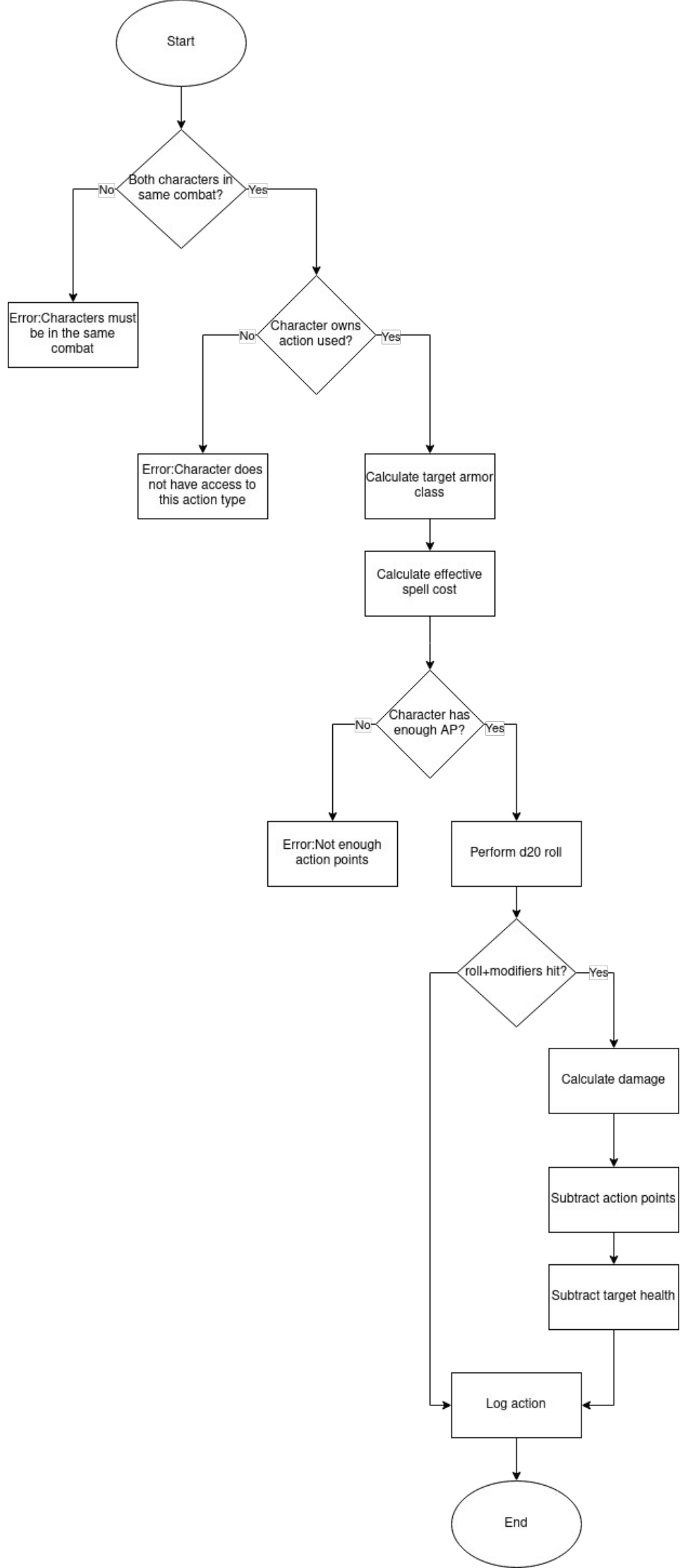
### enter\_combat:

- Parameters
  - combat\_id
  - character\_id
- Flow
  - Validate combat status
  - Validate character isn't in combat already
  - Update chracter action\_points to max and combat to combat\_num
  - Insert log action into action table



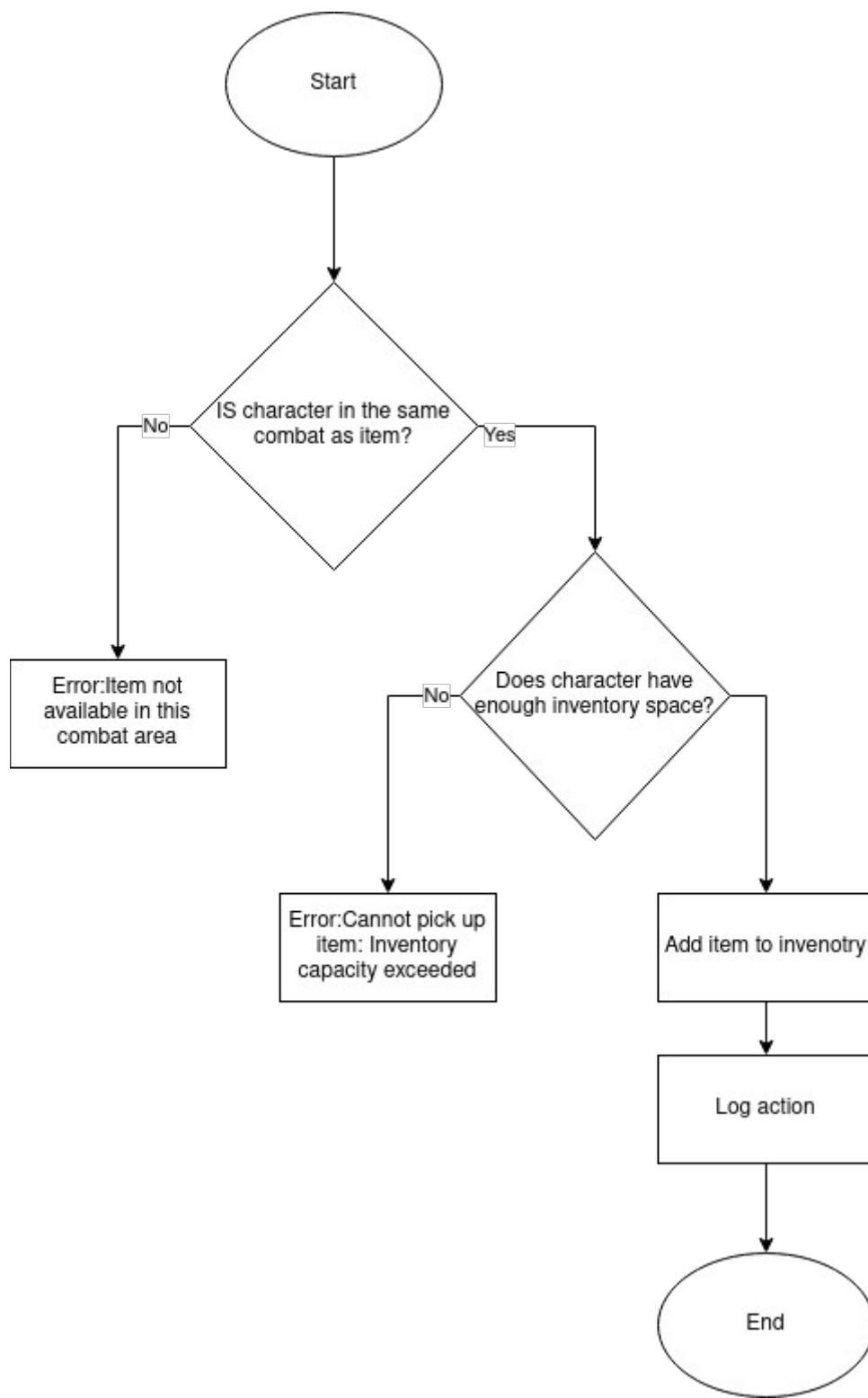
**run\_damage\_action:**

- Parameters
  - action\_type\_id
  - target\_id
  - character\_id
  - item\_id
- FlowValidate
  - Validate both characters are in the same combat
  - Validate character owns that action
  - Validate character has enough action points
  - Perform d20 roll adjusted by modifiers to calculate if character hit the target's
  - Log action
  - Deduct action points from caster
  - Deduct health from target
  - Log death if target died



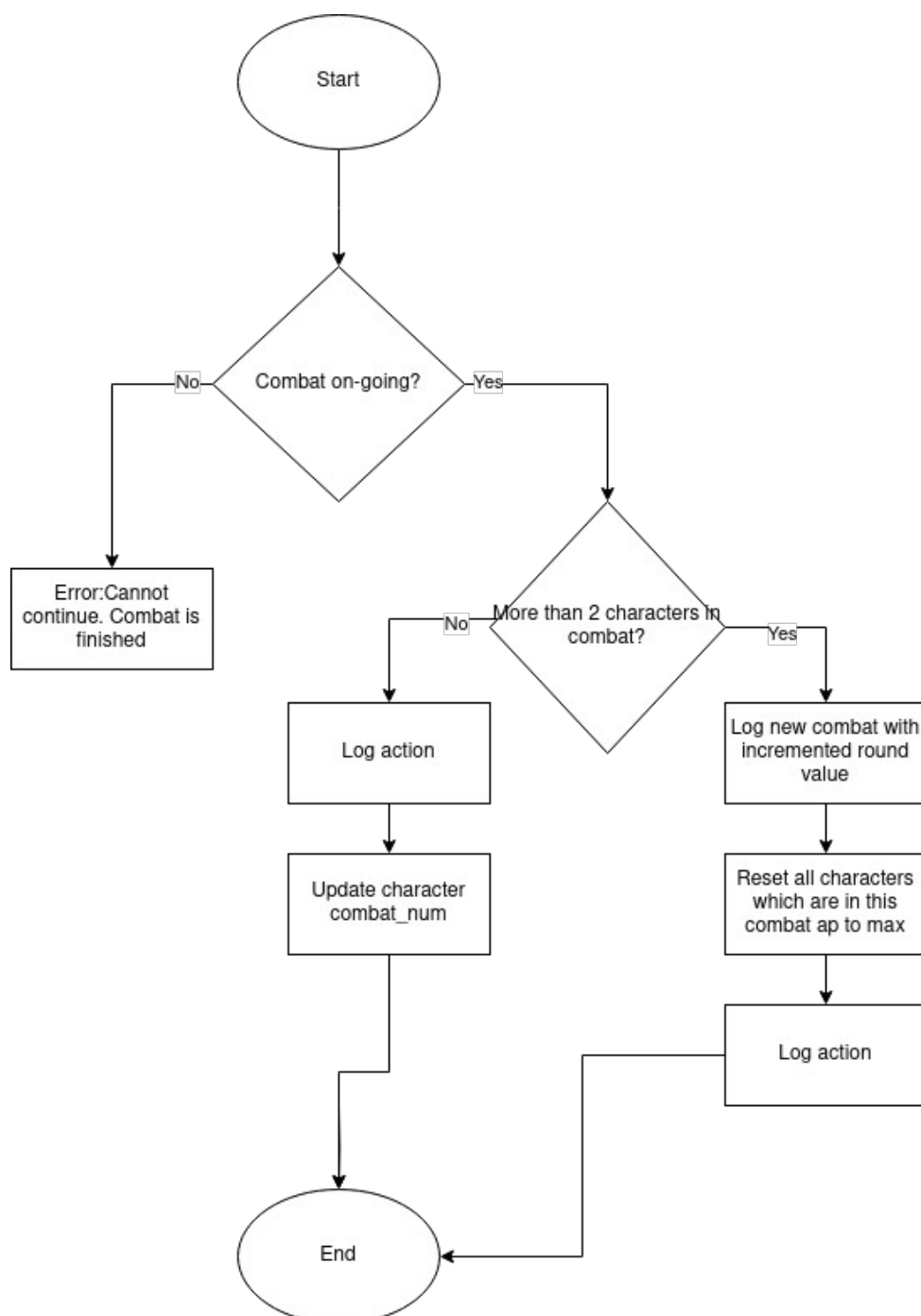
### loot\_item:

- Parameters
  - character\_id
  - item\_id
- FlowValidate
  - Confirm that character is in the same combat as item
  - Validate that character has enough inventory space to carry this item
  - Move item from playground to character inventory
  - Log the action



## reset\_round

- Parameters
  - combat\_id
- FlowValidate
  - Validate combat exists
  - Count number of participants
  - Finish combat if  $<2$  participants are in combat
  - Find highest round number and increment it by 1
  - Update combat table and character action points





## rest\_character

- Parameters
  - character\_id
- FlowValidate
  - Calculate max health
  - Confirm that character is not in combat
  - Execute healing action

